

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

**УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ЛЕСОТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ**

Кафедра информационных технологий и моделирования

О.А. Карасева

ПРОГРАММИРОВАНИЕ

VISUAL BASIC 2010

Часть 2

Учебно-методические указания по проведению лабораторно-практических
занятий для студентов

Направления 090303 (230700.62) – Прикладная информатика
дневной, очно-заочной и заочной формы обучения

ЕКАТЕРИНБУРГ

2015

Проект 1. Обработка баз данных с использованием технологии ADO.NET

Шаг 1. Создание базы данных SQL Server

Создайте папку для проекта. В базе данных будет одна таблица, содержащая сведения о телефонах ваших знакомых: Имя, Фамилия, Номер телефона.

1. Запускаем VB10, закажем новый проект *Windows Form Application* и укажем имя проекта `BD_sdf_2`.
2. Создаем новую базу данных.
3. В пункте меню *Проект* выбираем команду *Создать новый элемент*, выбираем шаблон *Локальная база данных*, указываем имя файла `Database1` (расширение будет `.sdf`).

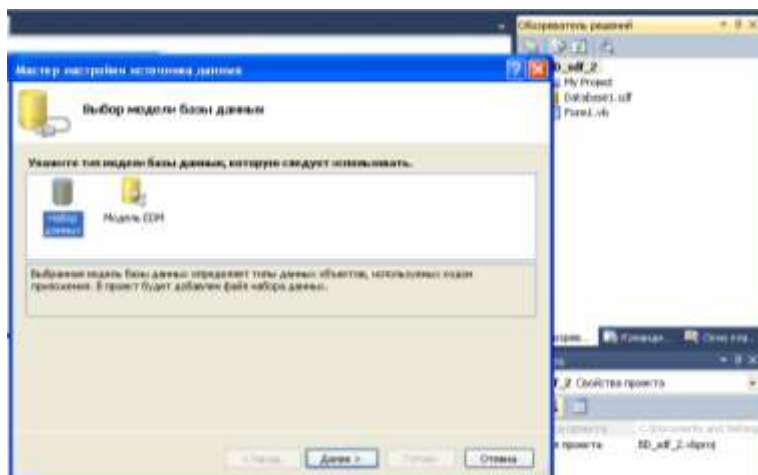


Рис.1

4. Добавляем таблицу в базу данных, для этого:
 - в меню *Вид* выбираем команду *Обозреватель баз данных* (*Обозреватель серверов*).
 - Разворачиваем узел *Database1* (рис 2), выбираем *Таблицы*.
 - Щелкаем правой кнопкой мыши пункт *Таблицы* и выбираем пункт *Создать Таблицу*.
 - Назовем таблицу БД телефонов. Заполним структуру таблицы, как показано на рис. 3.

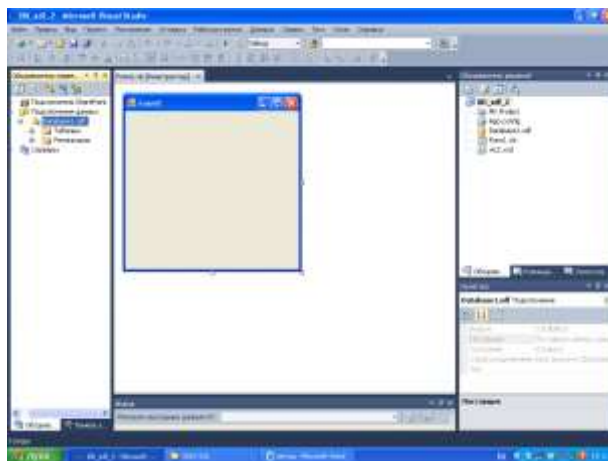


Рис.2

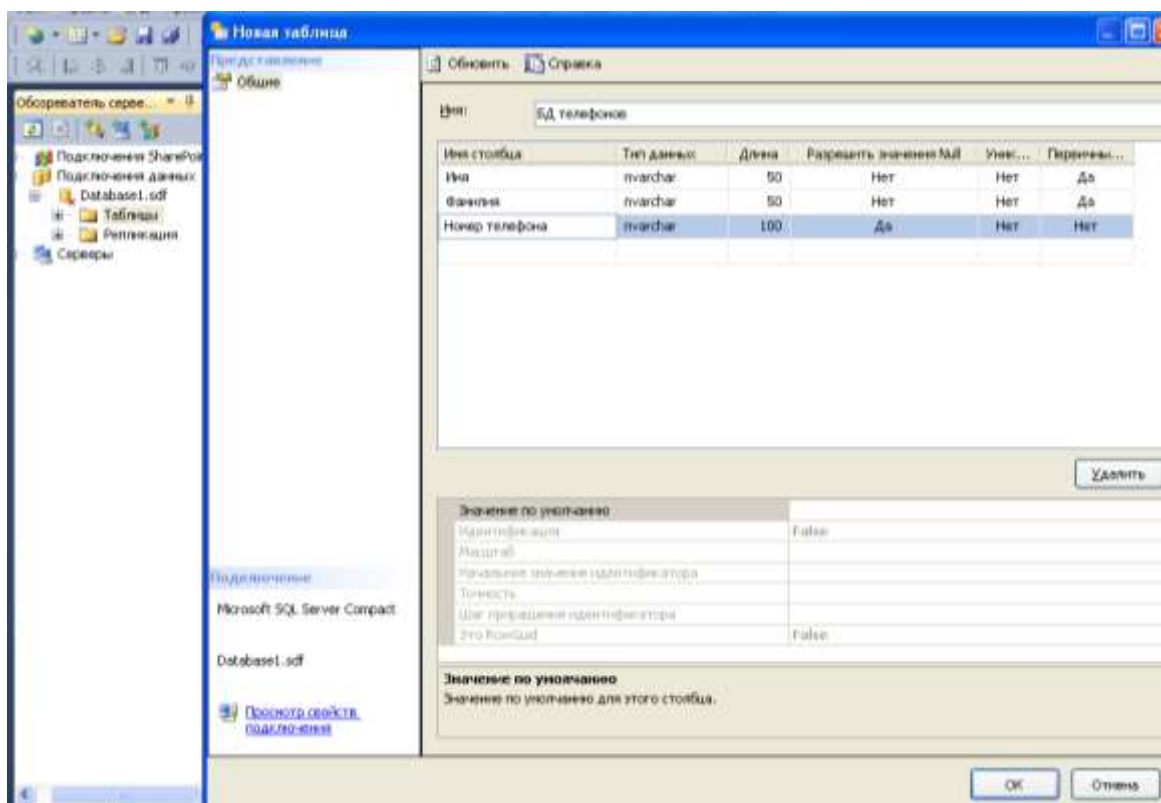


Рис.3

5. Добавим данные в таблицу. Для этого:

- В окне *Обозреватель Серверов* щелкнем правой кнопкой мыши на пункте БД телефонов и выберем команду *Показать Таблицу*. Откроется пустое окно данных таблицы.
- Заполним таблицу (рис. 4).
- После ввода выберем команду *Сохранить Все* для сохранения проекта и базы данных.

- Убедитесь, что в папке проекта появился файл с расширением .sdf. Его можно открыть вне проекта с помощью VB 2010 для редактирования базы данных.

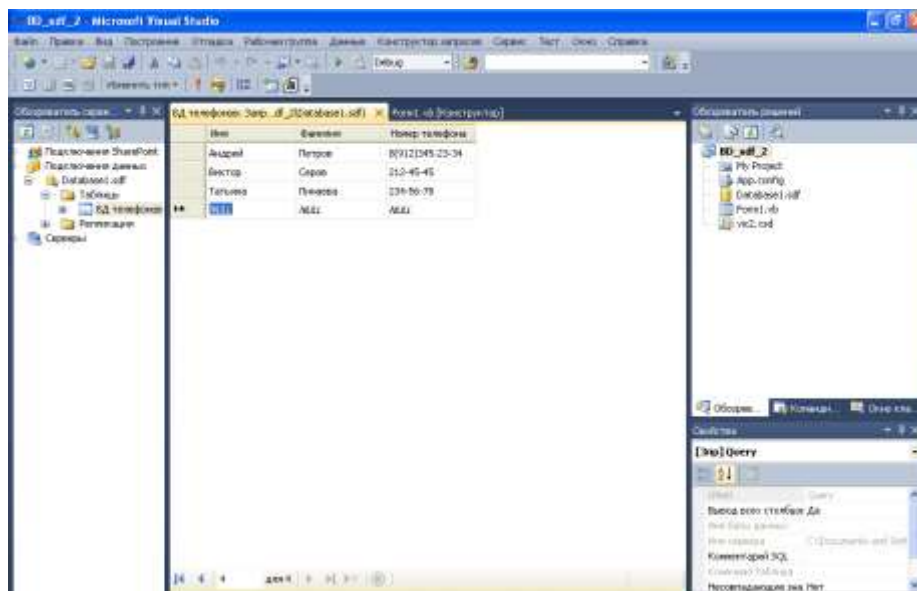


Рис.4

Шаг.2. Отображение таблицы базы данных SQL Server в экранной форме

Имея базу данных SQL Server, например, в виде файла Database1.sdf, можно легко вывести таблицу из этой базы данных в экранную форму.

1. Запустим VB10, закажем новый проект *Windows Application* и укажем имя проекта *BD_sdf_3*.
2. В меню *Вид* выбираем команду *Обозреватель баз данных* (Обозреватель серверов).
3. Щелкнув правой кнопкой мыши на значке *Подключение данных* (рис.5), добавим подключение. В качестве источника данных укажем *Microsoft SQL Server Compact 3.5., Продолжить*.
4. Далее нажмем кнопку *Обзор*, найдем созданный на предыдущем шаге файл базы данных *Database1.sdf*, далее *Открыть*, *OK*.

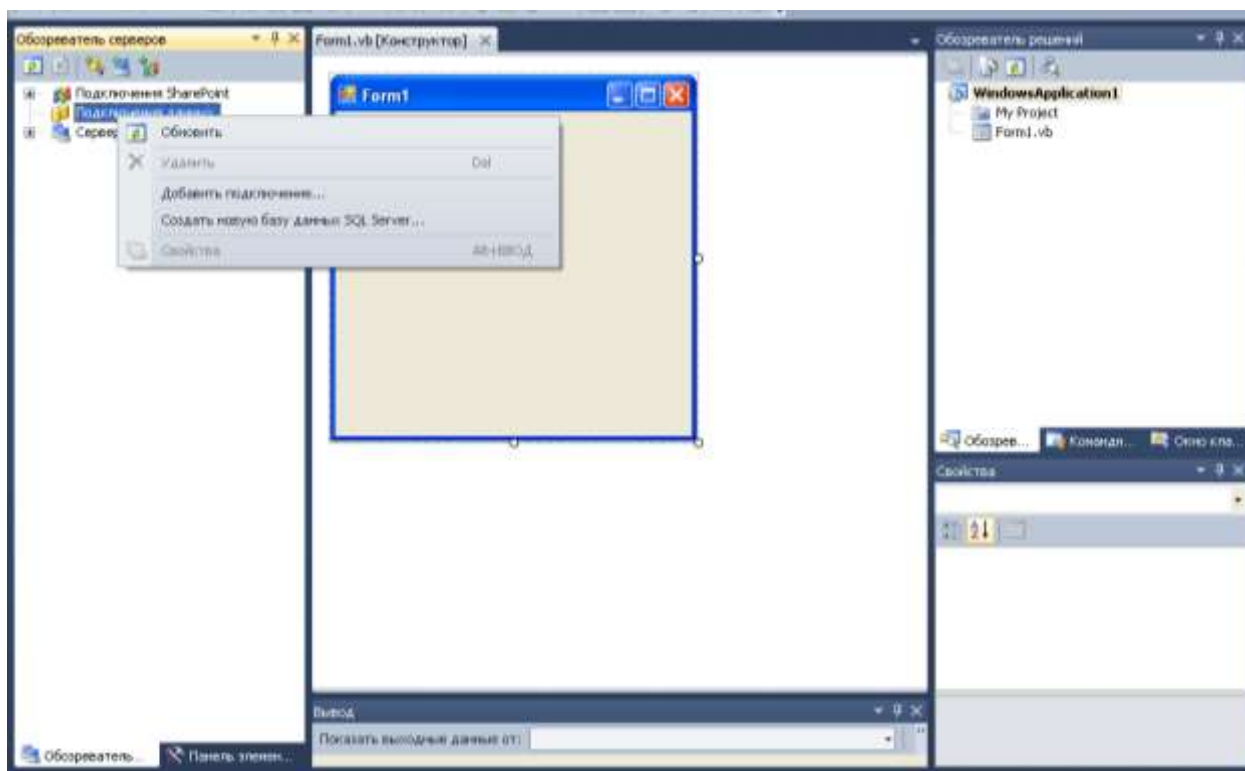


Рис. 5.

5. В пункте меню *Данные* выберем команду *Добавить новый источник данных*, запустим *Мастер настройки источника данных*. На запрос мастера, копировать ли файл в наш текущий проект, нажмем кнопку *Да* (рис.6-8).

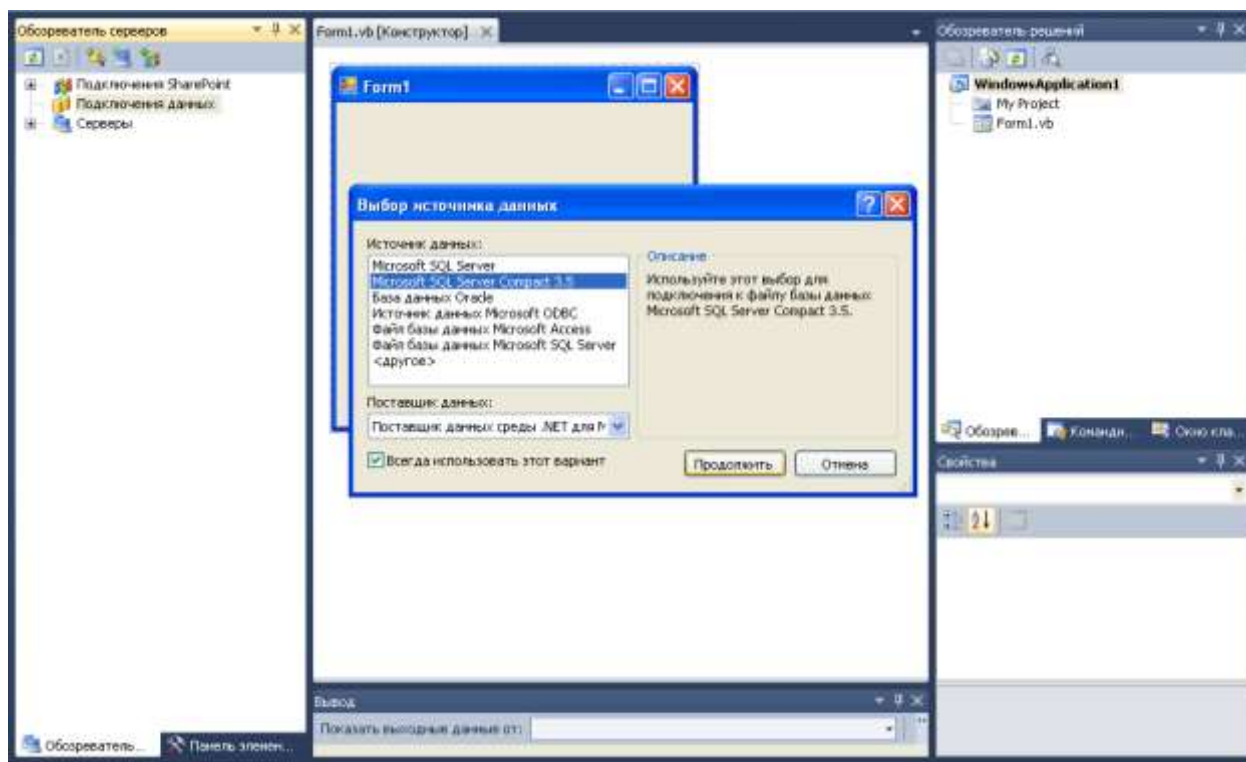


Рис.6.

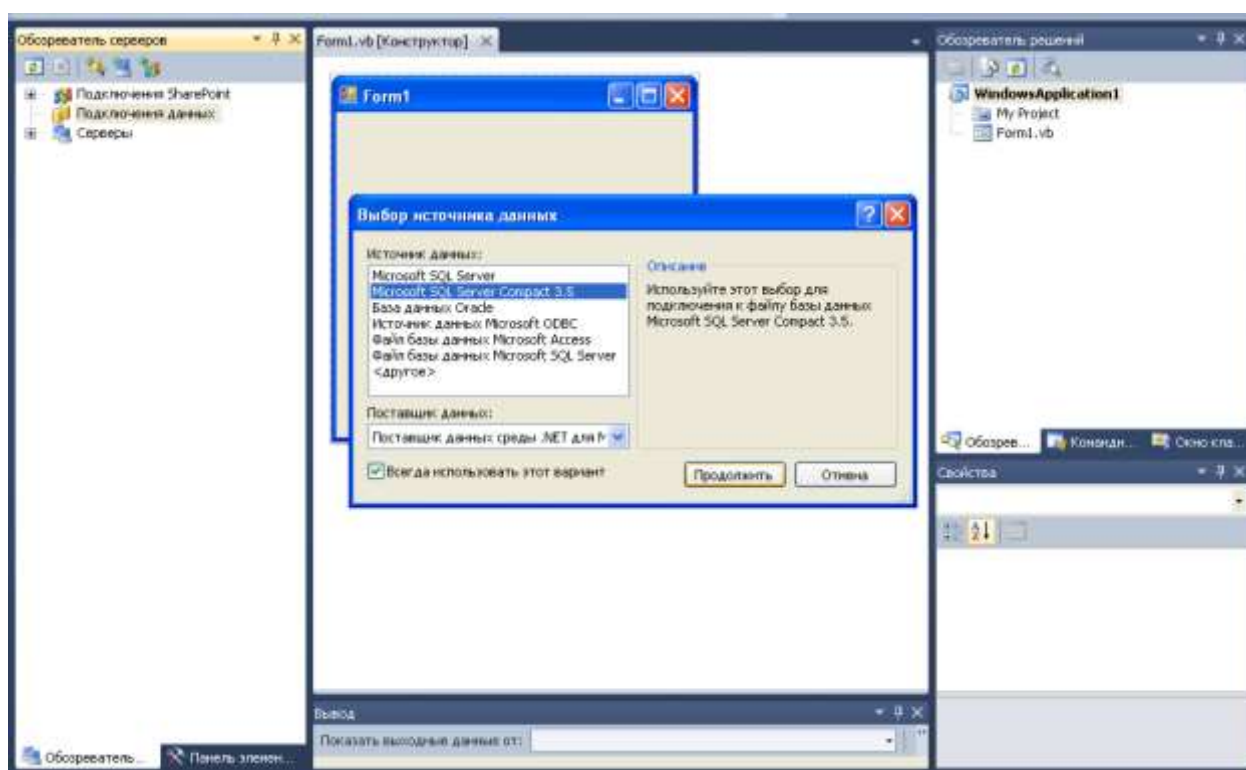


Рис 7.

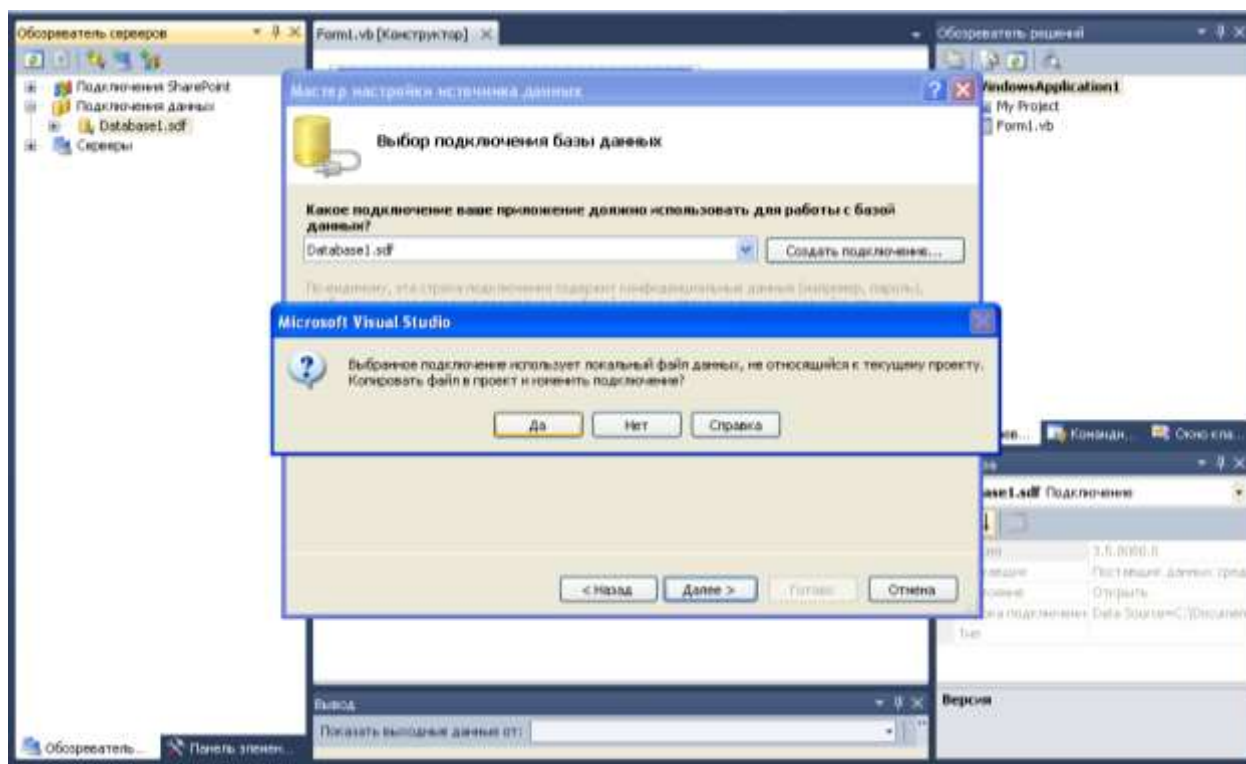


Рис.8

9. В результате в окне *Обозревателя решений* появится значок базы данных Database1.sdf и объекта Database1DataSet.xsd, но база данных в текущем проекте будет пустой. Для заполнения базы данных в пункте меню *Данные* укажем *Показать источники данных* и в появившемся окне *Источники данных* (рис.9-10), щелкая правой кнопкой мыши значок Database1.sdf, выберем пункт *Настроить источник данных с помощью мастера*.

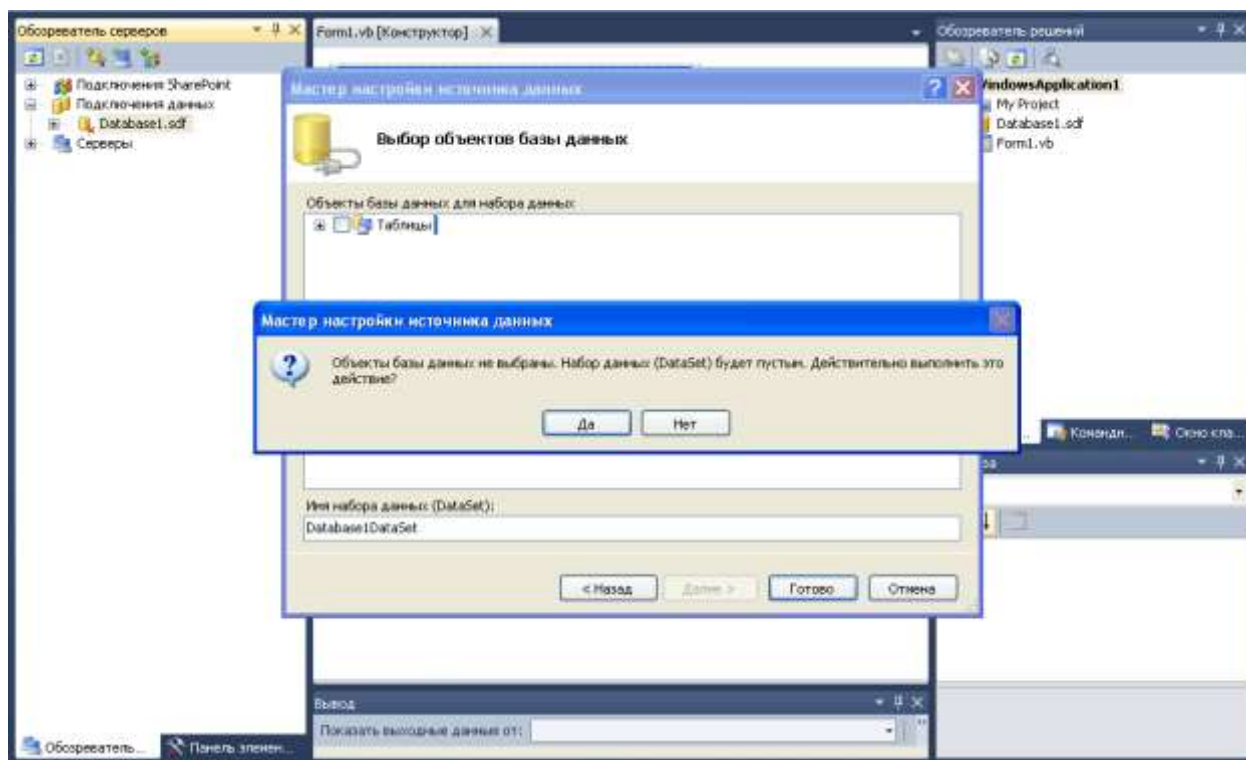


Рис. 9.

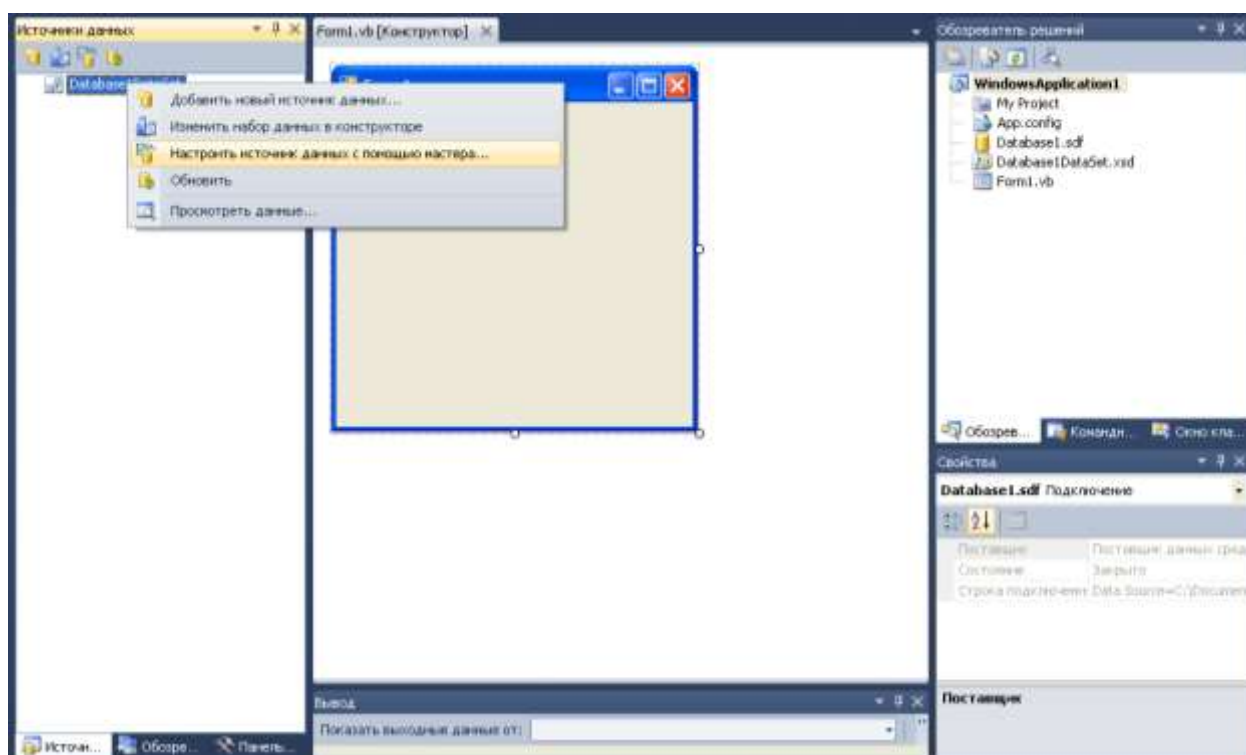


Рис.10

10. Теперь из окна *Обозревателя серверов* перетащим узел базы данных БД телефонов на появившуюся вкладку *Database1DataSet.xsd* (рис.11).

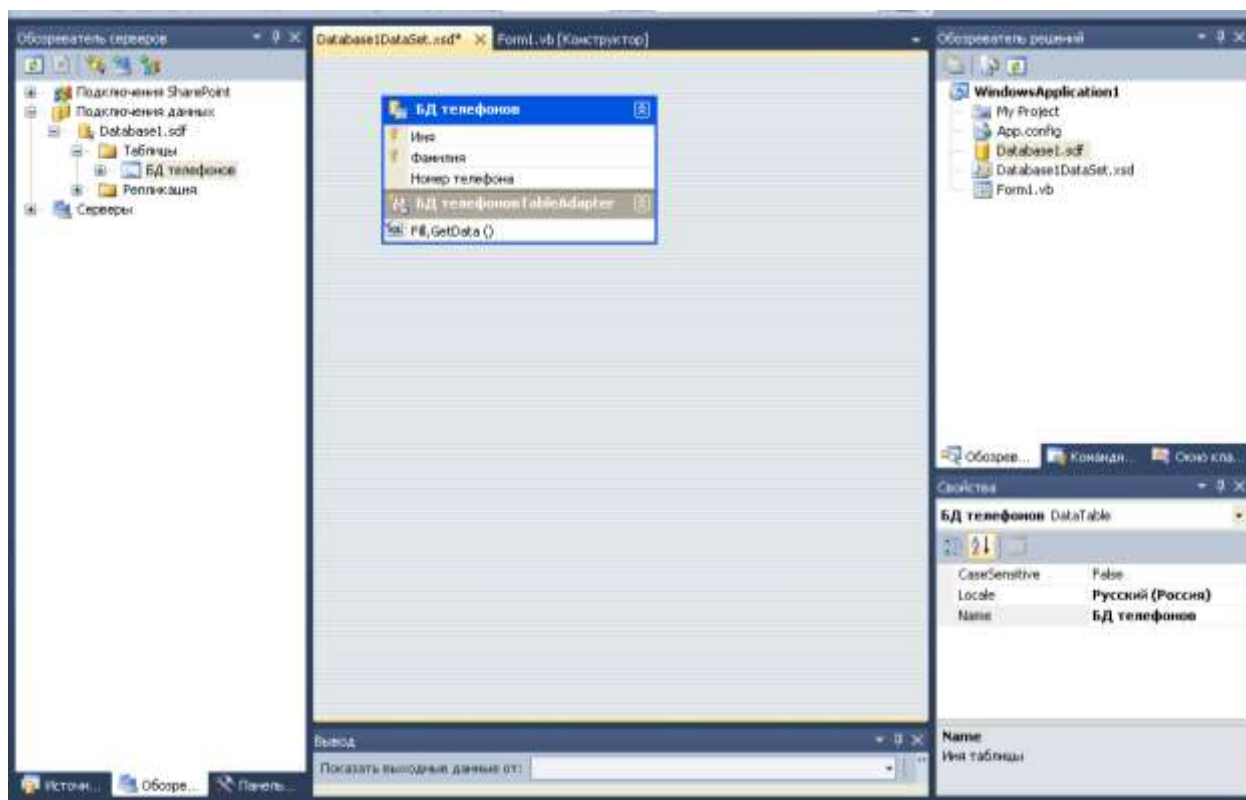


Рис.11

11. Из окна *Источники данных* перетащим узел базы данных БД телефонов на вкладку конструктора форм Form1.vb [Design] (рис.12). Как видно из рис. 12, автоматически появилось несколько программных объектов, необходимых для отображения данных из таблицы.

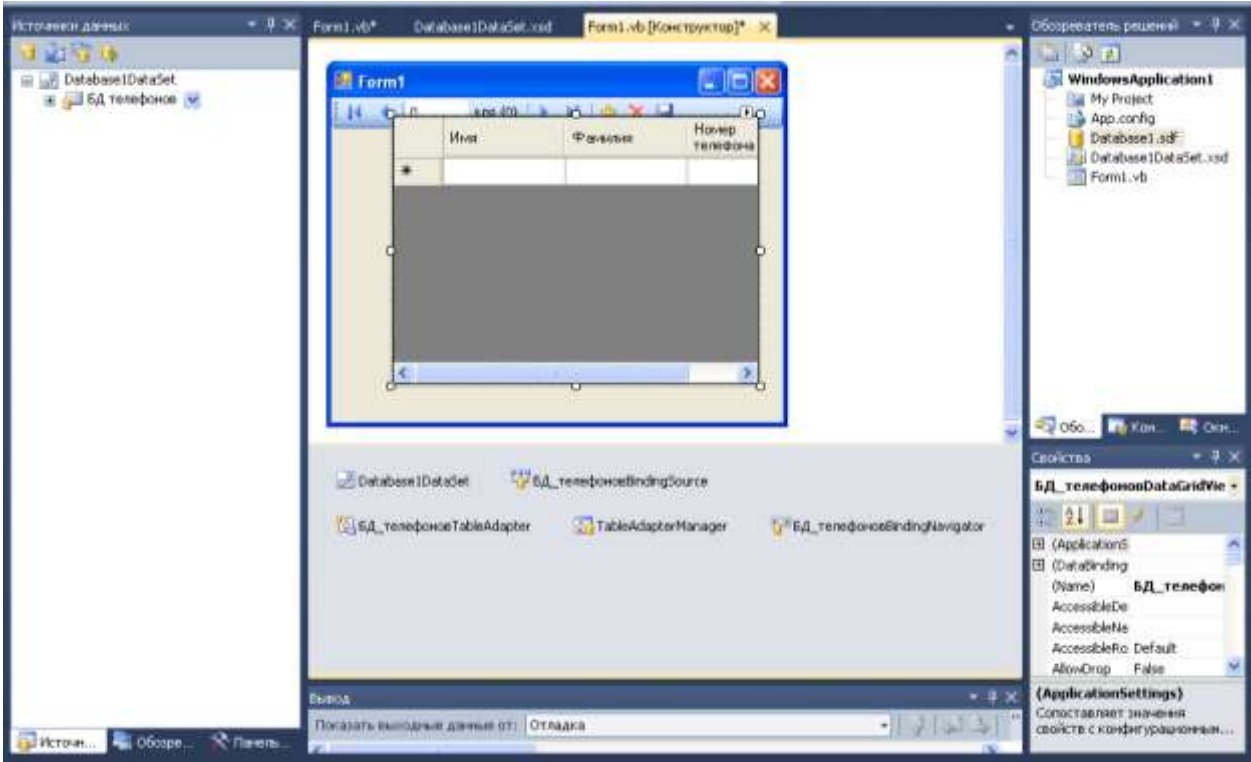


Рис.12. Проект формы с объектами для отображения таблицы БД

12.Стартуем отладку программы (клавиша F5). Благодаря элементам управления DataGridView и BindingNavigator появилось отображение таблицы из базы данных (рис.13). Мы имеем возможность модифицировать данные с помощью соответствующих кнопок.

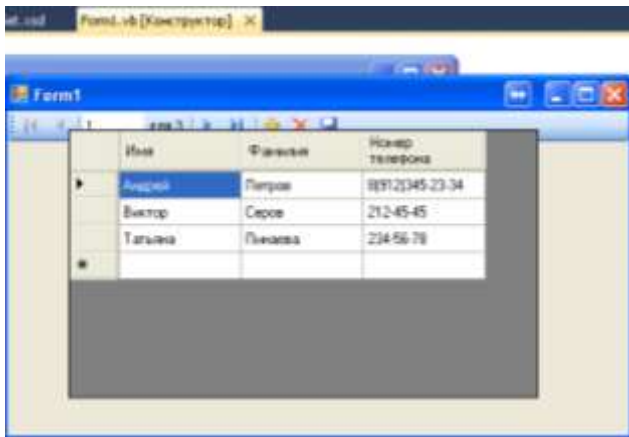
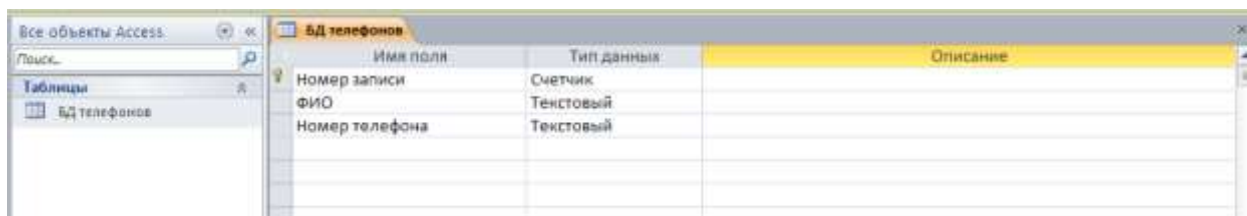


Рис. 13

Проект 2. Создание БД в среде MS Access и подключение к VB 2010

Шаг 1. Создание БД в среде MS Access

1. Создадим БД согласно структуре, представленной на рис.2.1 и назовем ее БД телефонов.



Имя поля	Тип данных	Описание
Номер записи	Счетчик	
ФИО	Текстовый	
Номер телефона	Текстовый	

Рис.2.1

2. Заполните базу данных несколькими записями.
3. Закроем БД Access и откроем таблицу БД телефонов в среде VB 2010.

Шаг 2. Редактирование таблицы в среде VB 2010 без написания программного кода

1. Запускаем VB 2010, но далее не заказывает новый проект.
2. Выбираем пункт меню *Вид, Обзорщик серверов*. Щелкнув правой кнопкой мыши на значке *Подключение данных* выберем пункт *Добавить подключение* и укажем в качестве источника данных, нажав кнопку *Изменить* (рис.2.2) *Файл базы данных MS Access* (рис.2.2).

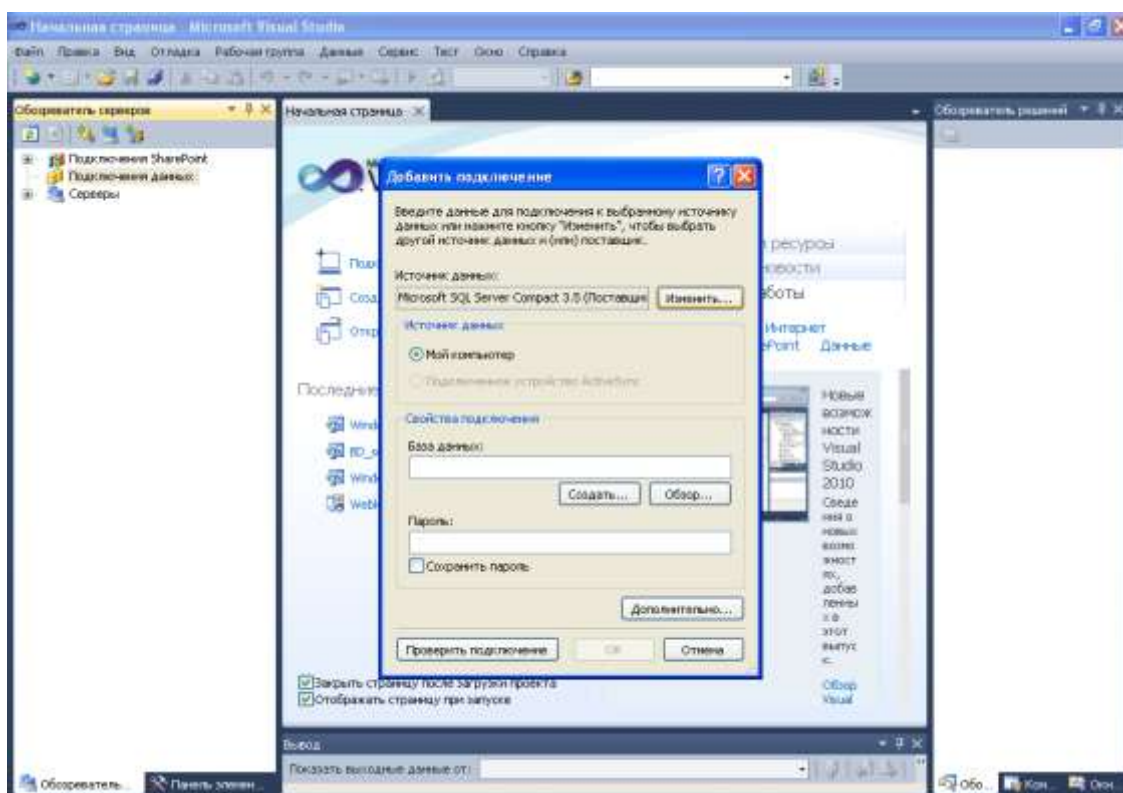


Рис.2.2

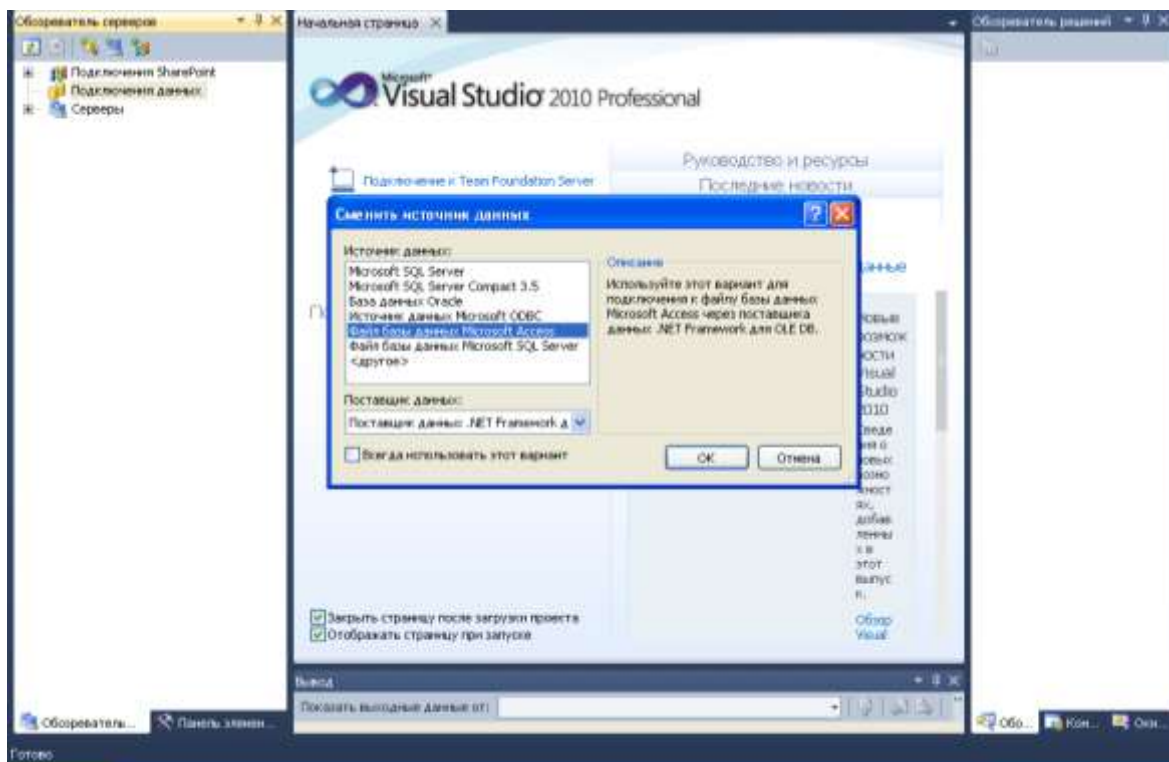


Рис.2.3.

3. Далее с помощью кнопки *Обзор* зададим путь и имя БД.
4. Проверим подключение, нажав кнопку *Проверить подключение*.
Успешное подключение выдаст сообщение, представленное на рис.2.4.

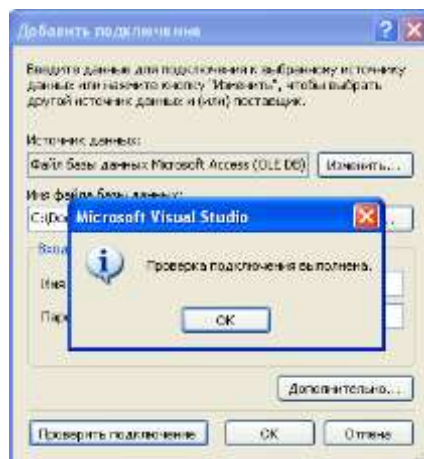


Рис.2.4.

5. Щелкаем по кнопке *OK*. В окне *Обозреватель серверов* раскроем узлы БД (рис.2.5).

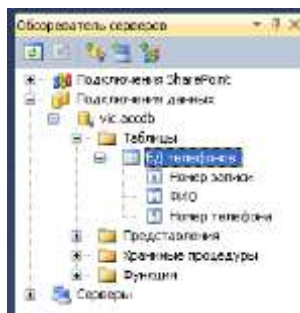


Рис.2.5.

6. Щелкаем правой кнопкой мыши на узле БД телефонов и выбираем команду *Загрузить данные*. В результате в правом окне получим содержимое таблицы с возможностью редактирования (рис.2.6).

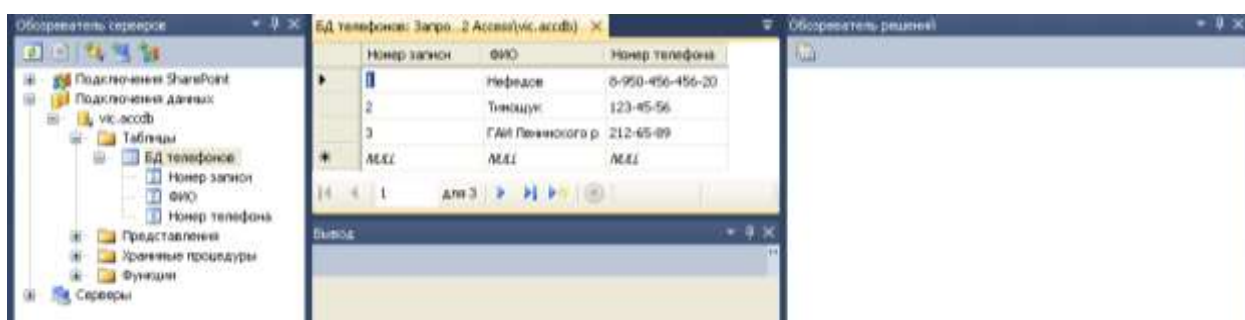


Рис.2.6

Шаг 3. Отображение таблицы в экранной форме

В предыдущем проекте (для базы данных SQL0 Server) мы копировали базу данных в наш проект. Теперь мы подключаемся к существующей базе данных.

6. Запускаем VB10, закажем новый проект *Windows Form Application* и укажем имя проекта Проект 2 .
7. В меню *Данные* щелкнем *Добавить новый источник данных*. При этом запустится *Мастер настройки источника данных*.
8. Выберем *База данных*, нажмем кнопку *Далее*, выберем *Набор данных*, нажмем кнопку *Далее*, кнопку *Создать подключение*.
9. В диалоговом окне *Добавить подключение* проверяем, установлен ли источник *Файл базы данных MS Access*, если нет, нажимаем кнопку *Изменить* и устанавливаем *Файл базы данных MS Access*, нажимаем кнопку *OK*.
10. Щелкнув по кнопке *Обзор*, выбираем место хранения базы данных vis.mdb, нажимаем кнопку *OK*.

11. После закрытия окна *Добавить подключение* в окне *Мастера настройки источников данных* нажмем кнопку *Далее*. Если будет предложено копировать файл данных в проект, нажмите кнопку *Нет*.
12. Нажмите кнопку *Далее*. На следующей странице мастера необходимо установить флажок у всех объектов базы данных (рис. 2.7). Имя объекта DataSet будет vicDataSet.

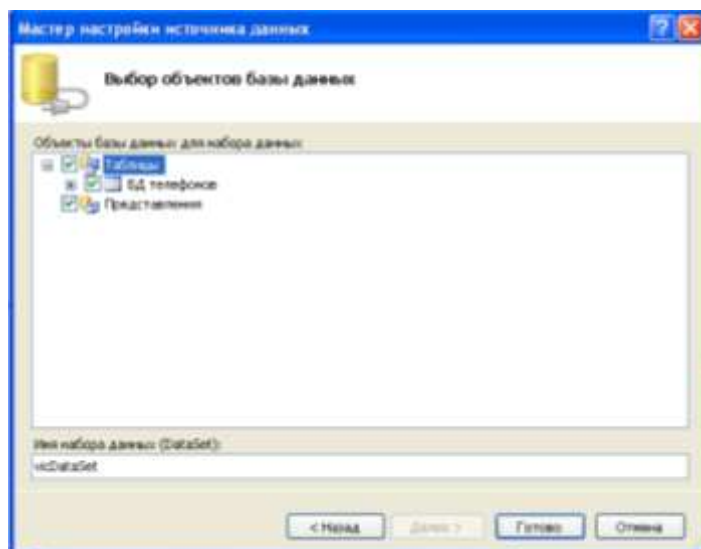


Рис.2.7

13. Нажимает кнопку *Готово*. Переходя в пункт меню *Данные, Показать источники данных* увидим в окне *Источники данных* узлы объектов базы (рис.2.8).



Рис.2.8

14. Перетащим узел БД телефонов из окна *Источники данных* в проект экранной формы (рис.2.9). При этом некоторые элементы управления автоматически добавляются в форму:

БД_телефоновBindingNavigator- элемент управления для переходов;

БД_телефоновBindingSource –подключение;

БД_телефоновTableAdapter - управление извлечением и обновлением данных;

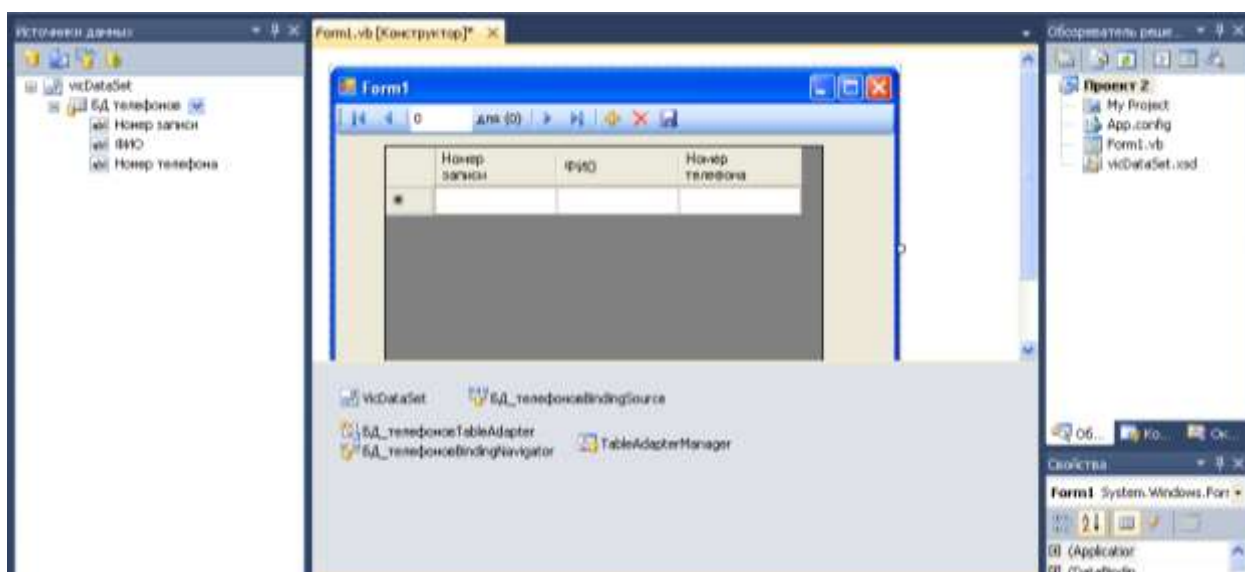


Рис. 2.9

15. Запустим приложение. Данные из таблицы отображаются в элементе управления DataGridView (рис.2.10).

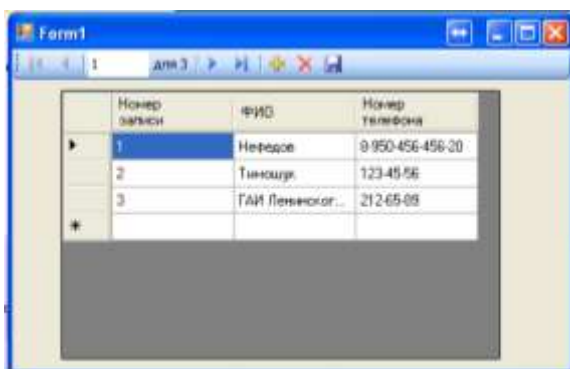


Рис.2.10

Можно использовать BindingNavigator в верхней части формы для перехода между строками. Можно изменять записи, но изменения будут сохранены только после нажатия кнопки сохранения.

Мы не написали код. Среда VB2010 автоматически сгенерировала программный код для управления программой.

Проект 3. Чтение записей из таблицы БД MS Access на консоль. Объекты Command, DataReader

При написании кода будет использована наиболее современная технология ADO.NET. Основные объекты этой технологии обеспечивают:

Connection –соединение с базой данных;

Command – привязку SQL-выражения к соединению с базой данных;

DataSet, DataReaders – просмотр результатов запроса.

Мы рассмотрим 4 основных действия над базой данных:

Select – выборка записей из таблицы;

Insert – вставка записей;

Update – модификация записей;

Delete – удаление записей.

Шаг 1. Создание проекта

1. Сохраните базу данных, созданную в предыдущем проекте в формате 2003 (Открыть, Файл, Сохранить и опубликовать, Сохранить как) под именем vic2003.
2. Запустим VB2010 и щелкнем на пункте Создать проект. Форма нам в данном случае не нужна, поэтому выбираем шаблон Console Application (Консольное приложение). Мы будем выводить таблицу из базы на консоль (на черный экран DOS). В этом случае будет использован моноширинный шрифт, следовательно. колонки в построенных таблицах будут ровными. Вид работающего приложения представлен на рис.3.1.

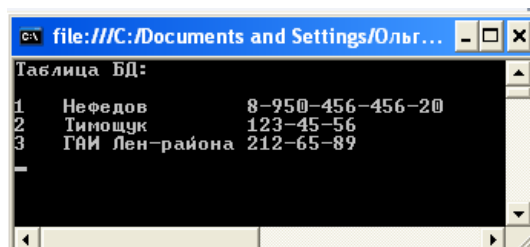


Рис.3.1

3. На вкладке Module1.vb напишем код:

```
Imports System.Data
Module Module1
```

```
Sub Main()
```

```
'Создаем объект Connection, передаем ему строку соединения
```

```
Dim Connection As New OleDb.OleDbConnection("Data
Source=""C:\Documents and Settings\Ольга\Рабочий стол\VB10-
SQL\Проект 3 Access+код\vic2003.mdb"";user " &
```

```
"ID=admin;Provider=""Microsoft.Jet.OLEDB.4.0"";")
```

```
'Создаем объект Command, передаем ему SQL-команду
```

```
Dim Command As New OleDb.OleDbCommand("select * from [БД
телефонов]", Connection)
```

```
'можно с сортировкой по фамилии:
```

```
'Dim Command As New OleDb.OleDbCommand("select * from [БД
телефонов] order by ФИО", Connection)
```

```
'Выполняем SQL-команду
```

```
Dim DataReader As OleDb.OleDbDataReader =
Command.ExecuteReader(CommandBehavior.CloseConnection)
```

```
Console.WriteLine("Таблица БД:" & vbCrLf)
```

```
While DataReader.Read() = True
```

```
'определяем количество полей в строке
```

```
'здесь три поля: 0,1 и 2
```

```
'минус прижимает строку влево.далее указана ширина поля
```

```
Console.WriteLine(String.Format("{0,-3} {1,-15}{2,-15}",
DataReader.GetValue(0), DataReader.GetValue(1), DataReader.GetValue(2)))
```

```
End While
```

```
DataReader.Close() : Connection.Close()
```

```
Console.Read() 'читает след. символ со стандартного ввода
```

```
End Sub
```

```
End Module
```

При работе DataReaders в памяти хранится только одна строка (запись) данных. Объект DataReaders имеет булеву функцию Read, которая возвращает True, если существует следующая запись, и False, если строки исчерпались. С помощью DataReaders можно заранее узнать количество записей в таблице.

Мы написали простейшую программу для просмотра записей таблицы базы данных. С ее помощью можно просматривать записи, но нельзя редактировать.

Проект 4. Создание БД Access. Создание структуры таблицы в БД. Добавление записей в таблицу

Шаг 1. Создание БД Access.

Программа работает следующим образом: создаем экземпляр класса ADOX.Catalog; его функция Create способна создавать БД, если на ее вход подать строку подключения; функция заключена в блоки Try... End Try, которые обрабатывают исключительные ситуации.

Технология работы

1. Создадим программу, которая во время работы создаст БД Access с именем new_BD.mdb. Эта БД будет пустой.
2. Запускаем VB2010, щелкаем пункт меню Создать проект и выбираем шаблон Консольное приложение.
3. Для добавления в наш проект DLL-библиотеки ADOX в меню Проект выберем команду Добавить ссылку; на вкладке COM дважды щелкнем на ссылке Microsoft ADO Ext.2.8 for DDL and Security.
4. Убедиться в том, что теперь существует ссылка на эту библиотеку, можно выбрав в меню Проект пункт Свойства консольного приложения и на вкладке Ссылки (рис.1) среди ссылок обнаружите ссылку Interop.ADOX.dll. Теперь можно ссылаться на это имя в программном коде, приведенном ниже.

Module Module1

```
Sub Main()  
    Dim cat As New ADOX.Catalog  
    Try  
        cat.Create("Provider=Microsoft.Jet.OLEDB.4.0;Data  
Source=E:\new_BD.mdb")  
        MessageBox.Show("База данных E:\new_BD.mdb успешно  
создана")  
    Catch Ex As Runtime.InteropServices.COMException  
        MessageBox.Show(Ex.Message)  
    Finally  
        cat = Nothing  
    End Try  
End Sub
```

End Try

End Sub

End Module

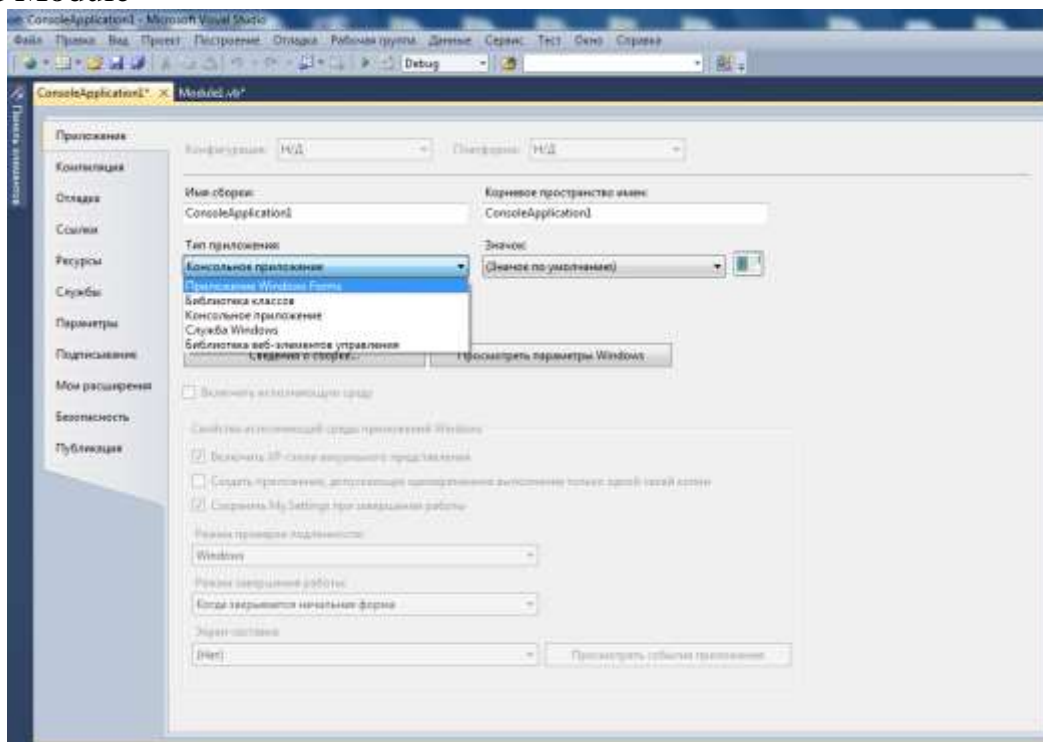


Рис 4.1

5. Далее, чтобы во время выполнения программы не появлялось черное окно DOS, в пунктах меню Проект, Свойства консольного приложения на вкладке Приложение в раскрывающемся списке Тип приложения выберем Windows Form Application.
6. Чтобы был доступен объект MessageBox, для вывода сообщений, добавим в проект еще одну DLL-библиотеку. Для этого выполните команды Проект, Добавить ссылку, и на вкладке .NET дважды щелкните ссылку System.Windows.Forms.dll, а в тексте выше строки Module Module1 напишите

Imports System.Windows.Forms

Ключевое слово Imports используется для импортирования пространства имен, которое содержит класс MessageBox

7. После запуска приложения получаем сообщение о создании БД (рис.4.2), при повторном запуске получаем соответствующее сообщение (рис.4.3). Данное сообщение генерировалось обработчиком исключительной ситуации.



Рис 4.2

Шаг 2. Создание структуры таблицы. Программная реализация подключения к БД

Здесь и далее мы будем использовать самую современную технологию ADO.NET. Для этого воспользуемся пустой БД, созданной на прошлом шаге.

Технология работы

1. Запускаем VB2010, щелкаем пункт меню Создать проект и выбираем шаблон Консольное приложение.
2. Пункты 3-6 предыдущего задания должны быть выполнены.
3. Наберите код:

```
Imports System.Data.OleDb
```

```
Imports System.Windows.Forms
```

```
Module Module1
```

```
    'запись структуры таблицы в пустую БД
```

```
    'моя пустая база данных new_bd_4-3.accdb хранится в корне диска
```

```
    F
```

```
    Sub Main()
```

```
        'создание экземпляра объекта connection с указанием строки  
        подключения
```

```
        Dim Connection = New
```

```
OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data  
Source=F:\new_bd_4-3.mdb")
```

```
        Connection.Open()
```

```
        'создание экземпляра Command с текстом SQL запроса
```

```
        Dim Command As New OleDbCommand("Create table [База  
телефонов] ([Номер п/п] counter, [ФИО] char(20),[Номер  
телефона]char(20))", Connection)
```

```
        Try
```

```
            Command.ExecuteNonQuery()
```

```
            MessageBox.Show("структура таблицы БД ТЕЛЕФОНОВ  
создана в пустой БД")
```

```
        Catch ex As Exception
```

```
            MessageBox.Show(ex.Message)
```

```
End Try  
Connection.Close()  
End Sub
```

```
End Module
```

4. Во время выполнения приложения вы получите сообщение (рис.4.3)

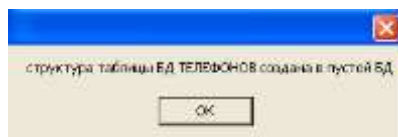


Рис 4.3

Шаг 3. Добавление записей в таблицу БД

Модифицируем всего лишь одну строку программного кода предыдущего задания для добавления новой записи в таблицу.

Обратите внимание на то, что имена таблиц следует точно указывать с учетом регистра. Однако, при наличии только одной таблицы это необязательно.

1. Создайте еще один модуль (Проект, Добавить модуль).
2. Скопируйте Module1 под именем Module2. Измените имя в коде.
3. Измените в свойствах консольного приложения значение поля Автоматически запускаемый объект на значение = modul2
4. Свойству Command задайте значение

```
Command.Connection = Connection
```

5. Программный код будет выглядеть так:

```
Imports System.Data.OleDb  
Imports System.Windows.Forms  
Module Module2  
    'добавление записи в таблицу БД  
    'моя база данных new_bd_4-3.mdb хранится в корне диска F  
    Sub Main()  
        'создание экземпляра объекта connection с указанием строки  
        подключения  
        Dim Connection = New  
        OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data  
        Source=F:\new_bd_4-3.mdb")  
        Connection.Open()
```

```
'создание экземпляра Command с текстом SQL запроса
Dim Command As New OleDbCommand("INSERT INTO [База
телефонов] ( [ФИО] ,[Номер телефона]) VALUES ('Ольга Карасева',
'111-222-333')")
' Для добавления записи в таблицу эта команда ОБЯЗАТЕЛЬНА!
Command.Connection = Connection
Command.ExecuteNonQuery()
MessageBox.Show(" В таблицу БД ТЕЛЕФОНОВ добавлена
запись ")

Connection.Close()
End Sub
```

End Module

6. Запустите приложение

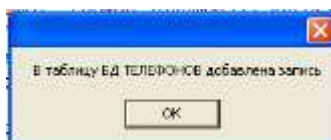


Рис 4.4

7. Чтобы проверить, появилась ли запись, необязательно открывать БД. Это можно сделать в среде VB. Для этого в пункте меню Вид выбираем команду Обозреватель серверов (Обозреватель баз данных), далее в списке подключений указываем полный путь к нужной БД. Затем щелкаем правой кнопкой мыши на значке нужной таблицы, в контекстном меню выбираем пункт Загрузить данные (рис. 4.5). Можно самим написать запрос (на панели инструментов Toolbar появляется значок SQL). Щелкнув по значку можно написать запрос, затем правой кнопкой мыши либо проверить синтаксис, либо выполнить запрос.

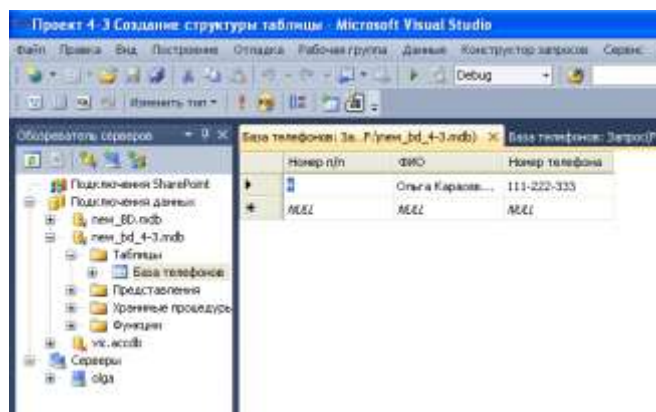


Рис 4.5

Проект 5. Чтение записей из таблицы. Объекты Command, DataReader, элемент управления DataGridView, Adapter и DataSet

Проект 5.1. Чтение данных с помощью объектов Command, DataReader, элемент управления DataGridView

Таблицу базы данных можно вывести в элемент управления DataGridView (сетка данных).

Порядок действий

1. Создать новый проект Проект 5-1 шаблона Windows Forms Application.
2. Из панели ToolBox добавить в форму элемент управления DataGridView и растянуть его на всю форму.
3. На событие загрузки формы Form1_Load написать программный код, представленный ниже.

```
Imports System.Data.OleDb  
Public Class Form1
```

```
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles MyBase.Load  
        Dim Connection As New OleDbConnection("Data  
Source=F:\new_bd_4-3.mdb;User  
Id=Admin;Provider=Microsoft.Jet.OLEDB.4.0;")  
        Connection.Open()  
        Dim Command As New System.Data.OleDb.OleDbCommand("Select *  
from [База телефонов]", Connection)  
        Dim DataREADER As OleDbDataReader = Command.ExecuteReader  
'(CommandBehavior.closeConnection)  
        Dim DATATable As New DataTable  
        'Заполнение шапки таблицы (три поля: 0,1,2)  
        DATATable.Columns.Add(DataREADER.GetName(0))  
        DATATable.Columns.Add(DataREADER.GetName(1))  
        DATATable.Columns.Add(DataREADER.GetName(2))  
        While DataREADER.Read() = True  
            DATATable.Rows.Add(New String() {DataREADER.GetValue(0),  
DataREADER.GetValue(1), DataREADER.GetValue(2)})  
        End While  
        DataREADER.Close() : Connection.Close()
```

```
DataGridView1.DataSource = DATATable
End Sub
End Class
```

4. Результат выполнения приложения представлен на рис.5.1. В этой таблице можно сортировать записи по любой из колонок, но результаты редактирования в базу не попадут, сохранения не будет. Преимущества объекта DataGridView – это его скорость и небольшое использование памяти. Однако при циклическом считывании данных эти преимущества исчезают.



Рис.5.1.

Проект 5.2. Чтение данных в сетку DataGridView с использованием объектов Command, Adapter и DataSet

Порядок действия

1. Создать новый проект Проект 5-2 шаблона Windows Forms Application.
2. Из панели ToolBox добавить в форму элемент управления DataGridView и растянуть его на всю форму.
3. На событие загрузки формы Form1_Load написать программный код, представленный ниже.

```
Imports System.Data.OleDb
Public Class Form1
```

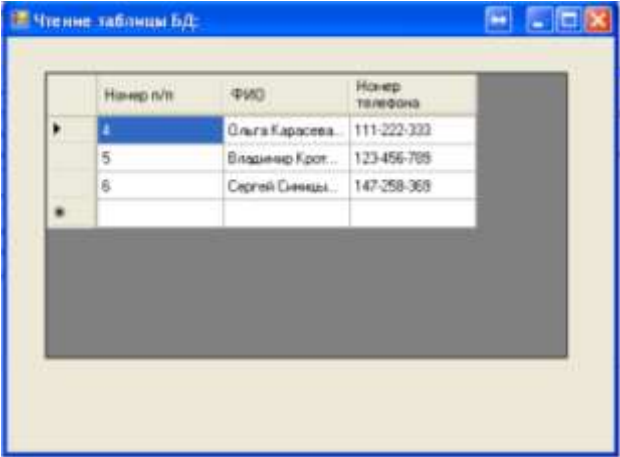
```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
Me.Text = "Чтение таблицы БД:"
```

```

Dim Connection As New OleDbConnection("Data
Source=F:\new_bd_4-3.mdb;User
Id=Admin;Provider=Microsoft.Jet.OLEDB.4.0;")
Connection.Open()
Dim Command As New System.Data.OleDb.OleDbCommand("Select *
from [База телефонов]", Connection)
Dim ADAPTER As New OleDbDataAdapter(Command)
Dim DATASET As New DataSet
ADAPTER.Fill(DATASET, "База телефонов")
DataGridView1.DataSource = DATASET
DataGridView1.DataMember = "База телефонов"
Connection.Close()
End Sub
End Class

```

4. Результат работы приложения представлен на рис. 5.2



Номер п/п	ФИО	Номер телефона
4	Ольга Карасева	111-222-333
5	Владимир Крот...	123-456-789
6	Сергей Сивид...	147-258-369

Рис 5.2

Также, как и при использовании объекта `DataReader` в предыдущем примере, в полученной таблице можно сортировать записи, изменять содержимое ячеек, но в базу данных эти изменения не попадут.

С помощью визуального проектирования выполнено только перетаскивание в форму сетки данных `DataGridView`, остальное сделано программно, что обеспечивает большую гибкость программы.

Проект 6. Модификация баз данных с использованием технологии ADO.NET

Проект 6-1. Обновление записей в таблице базы данных MS Access

Рассматриваемая в данном примере программа имеет форму, сетку данных DataGridView, в которую из базы данных читается таблица при нажатии кнопки Читать из БД (рис.5.1). Пользователь имеет возможность редактировать данные в этой таблице, после чего, нажав кнопку Сохранить в БД, данные в базе данных будут модифицированы, т.е. заменены новыми.

Порядок действий

1. Запускаем VB10, заказываем новый проект шаблона *Windows Form Application* и указываем имя проекта Проект 6-1.
2. Из панели инструментов добавляем в форму элемент управления DataGridView и две кнопки.
3. Программный код представлен ниже.

```
Imports System.Data.OleDb
```

```
Public Class Form1
```

```
    Private DS As DataSet
```

```
    Private DA As OleDbDataAdapter
```

```
    Private CO As New OleDb.OleDbConnection("Data  
Source=F:\new_BD.mdb;User Id=Admin;Provider=Microsoft.Jet.OLEDB.4.0;")
```

```
    Private CB As OleDb.OleDbCommandBuilder
```

```
    'Dim COMMAND = New OleDbCommand
```

```
    Dim COMMAND As OleDbCommand = New OleDbCommand
```

```
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles MyBase.Load
```

```
        Button1.Text() = "Читать из БД"
```

```
        Button2.Text() = "Сохранить в БД"
```

```
    End Sub
```

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Button1.Click
```

```
        DS = New DataSet
```

```
        If CO.State = ConnectionState.Closed Then CO.Open()
```

```
        DA = New OleDb.OleDbDataAdapter("select * from [База телефонов]",  
CO)
```

```
CB = New OleDb.OleDbCommandBuilder(DA)
DA.Fill(DS, "База телефонов")
DataGridView1.DataSource = DS.Tables("База телефонов")
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    COMMAND.CommandText = "Update [База телефонов] set [Номер
телефона]=?, ФИО=? where ([номер п/п]=?)"
    COMMAND.Parameters.Add("Номер телефона", OleDbType.VarWChar,
50, "номер телефона")
    COMMAND.Parameters.Add("ФИО", OleDbType.VarWChar, 50, "ФИО")
    COMMAND.Parameters.Add(New OleDbParameter("Original_Номер_п_п",
OleDbType.Integer, 0, System.Data.ParameterDirection.Input, False, CType(0,
Byte), CType(0, Byte), "номер п/п", System.Data.DataRowVersion.Original,
Nothing))
    DA.UpdateCommand = COMMAND
    COMMAND.Connection = CO
    Try
        'update возвращает количество измененных строк
        Dim kol As Integer = DA.Update(DS, "База телефонов")
        MessageBox.Show("обновлено " & kol & " записей")
    Catch ex As Exception
        MessageBox.Show(ex.Message, "недоразумение")
    End Try
End Sub
End Class
```

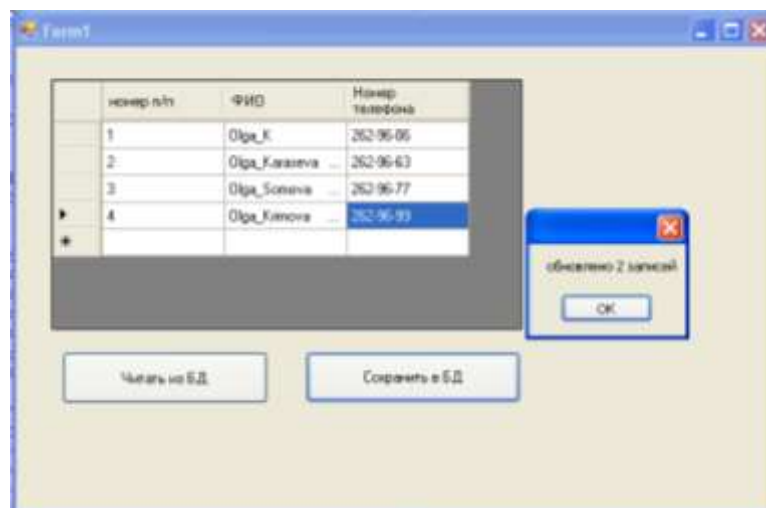


Рис.6.1

Поясним код.

Вначале импортируем (Imports) пространство имен System.Data.OleDb, это обеспечит более короткое объявление объектов и манипулирования ими. Далее, для того, чтобы объекты Dataset, DataAdapter, Connection, CommandBuilder и Command были видны во всех трех процедурах, объявляем их внешними внутри класса Form1.

При программировании чтения из БД на сетку DataGridView1 мы использовали CommandBuilder для демонстрации возможности чтения данных с помощью этого объекта, хотя могли бы организовать чтение из БД на сетку DataGridView1 так, как это было показано в предыдущей программе.

Рассмотрим процедуру сохранения записей в БД. Свойству CommandText присвоено значение текста SQL-запроса. В качестве заменителей параметров используются знаки вопроса. Эти параметры задаем с помощью метода COMMAND.Parameters.Add. Здесь указываем имя поля, тип, длину параметра и значение по умолчанию. Заметим, что третий параметр ("номер п/п") задается как новый, поскольку он не должен подлежать редактированию со стороны пользователя, а устанавливается автоматически.

Далее в блоке Try ...Catch вызываем непосредственно метод DA.Update, который возвращает количество обновленных записей. В случае неудачного

обновления обрабатывается исключительная ситуация Exception, объект Exception обеспечивает соответствующее сообщение.

Проект 6-2. Удаление записей из таблице базы данных с использование SQL-запроса и объекта Command

Удалять записи можно, формируя в программном коде соответствующий SQL-запрос, передаваемый в объект Command. Результат работы приложения – на рис. 6.2.

Порядок действий

1. Запускаем VB10, заказываем новый проект шаблона *Консольное приложение* и указываем имя проекта Проект 6-2.
2. Для того, чтобы во время выполнения программы не видеть черный экран DOS, в меню Проект выберем команду Свойства своего консольного приложения и на вкладке Приложение укажем в раскрывающемся списке Тип приложения тип приложения Приложение Windows Form.
3. Чтобы иметь доступ к функции MessageBox.Show (хотя мы можем использовать MsgBox), добавим к проекту ссылку на динамическую библиотеку Forms.dll. Для этого выберем пункты меню Проект, Добавить ссылку и на вкладке .NET дважды щелкнем на ссылке на библиотеку System.Windows.Forms.dll. Отметим, что на вкладке Проект, Свойства, Ссылки появилась соответствующая этой библиотеке ссылка.
4. Программный код представлен ниже.

```
Imports System.Data.OleDb
```

```
Imports System.Windows.Forms
```

```
Module Module1
```

```
    ' программа удаляет запись из таблицы БД с использованием SQL - запроса
```

```
    Sub Main()
```

```
        ' создаем объект Connection и передаем ему строку соединения
```

```
        Dim Connection As New OleDb.OleDbConnection("Data
```

```
Source=F:\new_BD.mdb;User Id=Admin;Provider=Microsoft.Jet.OLEDB.4.0;")
```

```
        Connection.Open()
```



```
Dim Command As New System.Data.OleDb.OleDbCommand("Delete * from  
[База телефонов] where ФИО like 'ОІ%'", Connection)  
  
Dim i As Integer = Command.ExecuteNonQuery  
  
' i- количество удаленных записей  
  
If i > 0 Then MessageBox.Show("записи, содержащие в поле ФИО  
фрагмент 'ОІ*', удалены")  
  
If i = 0 Then MessageBox.Show("запись, содержащая в поле ФИО  
фрагмент 'ОІ*', не найдена")  
  
Connection.Close()  
  
End Sub  
  
End Module
```



Рис. 6.2.

Здесь при создании объекта Command задан SQL –запрос на удаление всех записей, содержащих в поле ФИО фрагмент текста «ОІ*», причем строчные и прописные буквы являются равнозначными. Поиск записей ведется без учета регистра (case-insensitive search).

Для выполнения команды SQL использован метод ExecuteNonQuery. Он возвращает в переменную i количество удаленных записей.